UNIVERSITY OF TORONTO SCARBOROUGH

Assignment: 3
Course: CSCC43
Deadline: Nov 24th, 11:59 pm
Submission: via Quercus

This assignment that makes up 15% of your total mark must be done individually. For this assignment you will complete your software, that you have started to implement for your first assignment, by embedding SQL DDLs and DMLs into the code. However, before you do that, we want to make sure that everybody uses the same relations and attributes. Therefore, you need to follow the design that is provided in the last page of this text, which is a solution to the second assignment.
Implement the relations and establish the relationships by defining the foreign keys of the relations. Either insert data directly into the tables as much as you think it is enough to test the queries, or act smartly and insert data using *insert* menu of your software. So not only you insert data but also you test your code. You do **not** need to submit the script by which you create the tables since you have been assessed for this by the second assignment.

If you have a question that may reveal the solution, please post it privately in Piazza so that either me or my TAs would answer you.

If you have completed the first assignment, you only need to write SQL commands and embed them in your code to finish the project:

A. Insert
   This is where you populate your relations. Remember that, for example for a book, you get all the information about it in one page but you insert this data in multiple relations.

B. Update
   As was explained in the first assignment, here is where you get a name (i.e. a book, album or movie name), show all the information about it and let users edit it.

C. Remove
   By this, you can remove data about the item whose name was entered.

D. View
   For this part you only follow the instruction given in the first assignment and write a dynamic query. There is a safe and unsafe way of writing such queries. An unsafe way will be prone to SQL injection attack, which obviously you do not want to create such a code. To avoid this, you use SQL parameters. I will talk about it in lecture 12 but in the meantime here is a clear explanation on how to write a safe query: https://www.w3schools.com/sql/sql_injection.asp.
   Use of views and multiple *select* statements is allowed.

E. Report
   For this part you need to write 10 reports (i.e. SQL DMLs) and replace R1, R2, …R10 in the interface with the names that are given below. Use of views and multiple *select* statements is allowed. The results of the following queries should be sorted ascendingly unless otherwise is stated.

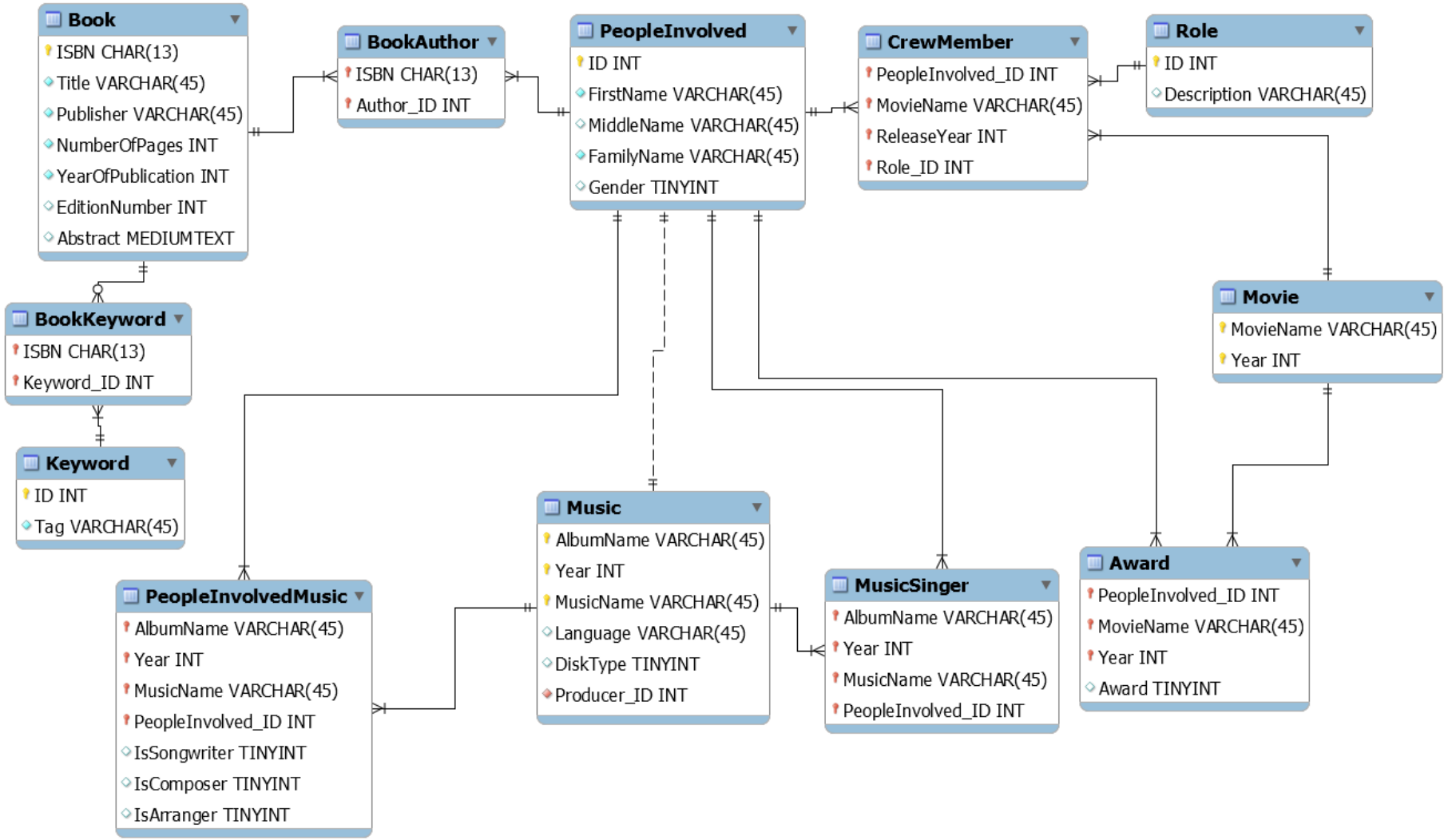
R	Report Name	Description	Input	Output	Order by
1	Authors' Publications	Find all the books that's been written by one author.	Author's Name	ISBN, Book's Name, Published year	ISBN
2	Publications in one Year	Find all the books, which published in the same year	Year	ISBN, Book's Name, Published year Authors' Family Name and Initial of First Name	Book's name
3	Books with Similar Topic	Find all the books, which is about a specific subject. The subject of a book can be presented by its title, description or keywords that define the book.	A string representing the subject	ISBN, Book's Name, Published year	ISBN
4	Frequent Publishers	Find all the authors who published books in at least two consecutive years.	--	ISBN, Book's Name, Author's name Published year	Author's name, year
5	Most Popular Subjects	Find the most popular subject in your library, which is the subject for which a similar tag was used more than once.	--	Tag, Frequency	Tag
6	Multi Skills Movie Crew	Find all the people who played more than one role in producing a movie.	--	Family Name, Role, Movie Name	Family Name
7	Award Winning Movies	Find all the movies directed by the same person and received at least one award.	---	Movie Name, Director Name, # of Awards	Director Name, Movie Name
8	Music with Similar Name	Find the singers of all the music, which share the same name.	---	Album Name, Music name, Singers, Year	Music Name, Year
9	Multi Skills Music Crew	Find all the people who are both the composer and song writer for a music but not are not an arranger.	---	Family Name, First Name Initial, Music Name, Album Name, Year	Year, Music Name Descending.
10	Similar Names	Find the similar family names that have been an author or part of music or movie production. It is possible that they are not the same person and only share the same family name.	--	Family name, Roles	Family Name

Submission: [This is the same as what is said in the first assignment**]**

- You only submit one code, which is HL_YourUTORid.java (e.g. HL_abc123.java). This file will be marked for your first and third assignment.
- When you complete the code for the third assignment, you should write a comment with the following format that shows which part of the code solves which question. You need to write this comment as close as possible to your SQL code. Here is the format:
  /* --- problem name --- */
  Examples:
  /* --- report 1: Authors' Publications --- */
  OR
  /* --- Data->insert->book ---*/

Marking Scheme:

- You will only be marked based on the correctness, safety and efficiency of your query. By efficiency, I mean how fast your query runs. For example, if a report can be produced by one single SQL *select* statement, I don't expect you to write more than one. Also, a non-correlated subqueries are preferred over correlated ones. Make sure that if you create a view, you create and use it wisely. It means that you do not create a view just because it makes the problem solving easier for you. Create a view, if it makes sense performance wise.


Book
ISBN CHAR(13)
Title VARCHAR(45)
Publisher VARCHAR(45)
NumberOfPages INT
YearOfPublication INT
EditionNumber INT
Abstract MEDIUMTEXT
BookAuthor
ISBN CHAR(13)
Author_ID INT
PeopleInvolved
ID INT
FirstName VARCHAR(45)
MiddleName VARCHAR(45)
FamilyName VARCHAR(45)
Gender TINYINT
CrewMember
PeopleInvolved_ID INT
MovieName VARCHAR(45)
ReleaseYear INT
Role_ID INT
Role
ID INT
Description VARCHAR(45)
Movie
MovieName VARCHAR(45)
Year INT
BookKeyword
ISBN CHAR(13)
Keyword_ID INT
Keyword
ID INT
Tag VARCHAR(45)
Music
AlbumName VARCHAR(45)
Year INT
MusicName VARCHAR(45)
Language VARCHAR(45)
DiskType TINYINT
Producer_ID INT
MusicSinger
AlbumName VARCHAR(45)
Year INT
MusicName VARCHAR(45)
PeopleInvolved_ID INT
Award
PeopleInvolved_ID INT
MovieName VARCHAR(45)
Year INT
Award TINYINT
PeopleInvolvedMusic
AlbumName VARCHAR(45)
Year INT
MusicName VARCHAR(45)
PeopleInvolved_ID INT
IsSongwriter TINYINT
IsComposer TINYINT
IsArranger TINYINT